

Funzioni API di base di Cisco Meeting Server

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Richiesta e risposta API](#)

[Configurazione](#)

[CMS 2.9 e versioni successive](#)

[Configurare un oggetto API](#)

[Modificare un oggetto API già creato](#)

[Eliminare un oggetto API dal menu API](#)

[Spiegazione delle richieste API \(utilizzando strumenti di terze parti\)](#)

[POST HTTP](#)

[HTTP GET](#)

[HTTP PUT](#)

[ELIMINAZIONE HTTP](#)

[Client API](#)

[POSTER](#)

[POSTINO](#)

[Verifica](#)

[Risoluzione dei problemi](#)

Introduzione

Questo documento descrive le quattro funzioni API di base (Application Program Interface) GET, POST, PUT, DELETE utilizzate in CMS (Cisco Meeting Server). A partire dalla versione CMS 2.9, l'interfaccia grafica dell'amministratore Web dispone di un menu API disponibile nel menu Configuration. Questo documento esamina il nuovo menu e descrive anche due diversi strumenti API: Poster e Postman e come utilizzarli per la configurazione CMS.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni di questo documento possono essere usate con CMS 2.9 e versioni successive o

con diversi client API come Postman o Poster. Questi strumenti di terze parti sono descritti nella sezione Client API del documento.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

L'API per il CMS è un modo estremamente flessibile per configurare molte delle sue funzioni. Ci sono troppe funzionalità API da memorizzare o da coprire, quindi assicuratevi di fare riferimento alla documentazione di riferimento dell'API corrente. Al momento della stesura di questo documento, [qui](#) sono disponibili le guide API Reference correnti.

Richiesta e risposta API

La comunicazione API è una relazione richiesta-risposta tra client e server. Il client effettua una richiesta del server. Dopo la gestione della richiesta (completamento o rifiuto di un'azione) viene restituita una risposta.

API Request and Response



Le quattro richieste descritte in questo articolo sono:

1. GET - Recupera le informazioni esistenti
2. POST - Crea nuove informazioni
3. PUT - Modifica le informazioni esistenti
4. DELETE - Elimina le informazioni esistenti.

Queste sono le richieste API di base utilizzate per configurare CMS.

La risposta più comune è un 200 OK. Altre risposte sono 4xx e 5xx, che sono risposte di errore.

Configurazione

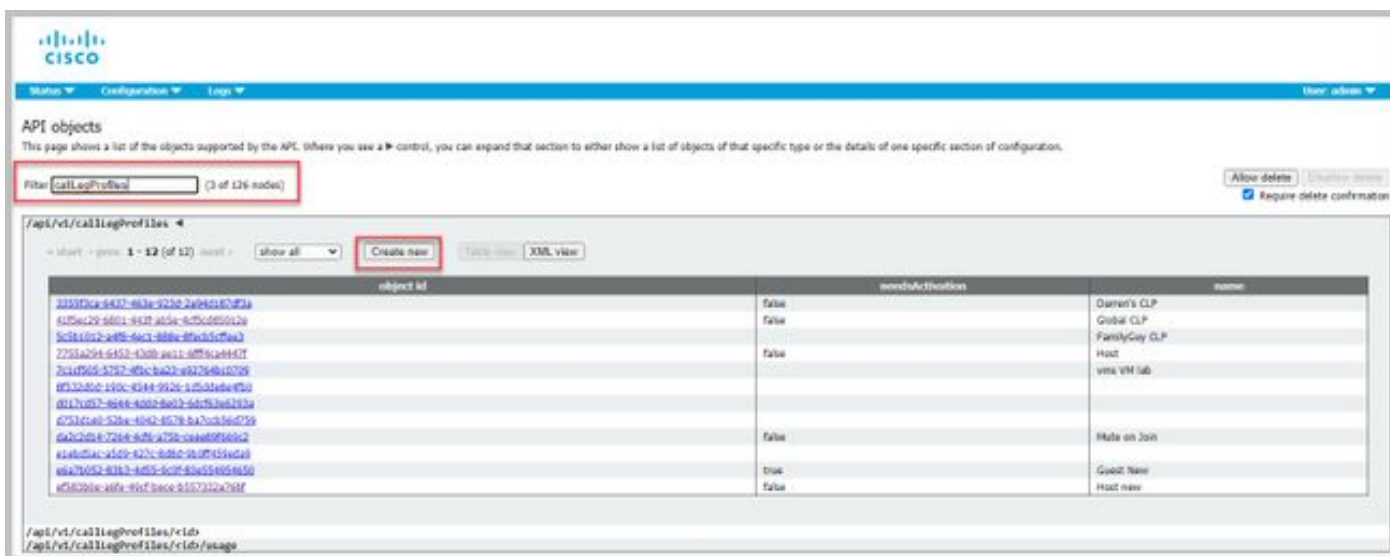
CMS 2.9 e versioni successive

CMS 2.9 ha introdotto un nuovo menu API che semplifica notevolmente la modifica delle impostazioni e la messa a punto delle impostazioni in CMS. Quando si utilizza il menu, vengono visualizzati tutti i parametri disponibili, il che consente di modificare le impostazioni in modo rapido e semplice e di attivare nuove funzioni.

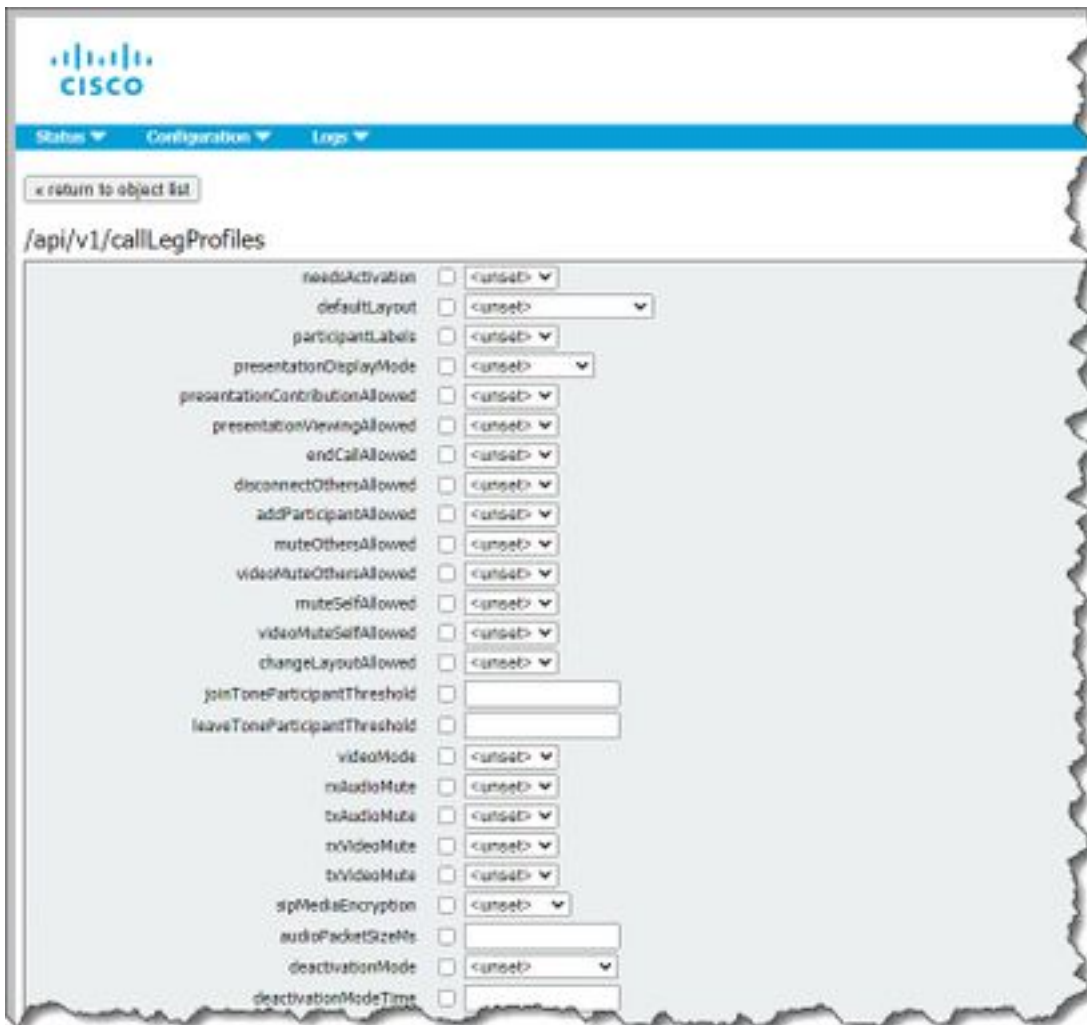


Configurare un oggetto API

Una volta nel menu API, è possibile filtrare gli oggetti API per quello che si sta cercando di modificare/creare e quindi fare clic sulla freccia nera accanto all'oggetto per apportare le modifiche. In questo esempio viene illustrato come cercare callLegProfiles e creare un nuovo callLegProfile.



Quando si fa clic su **Crea nuovo**, viene visualizzata questa schermata che mostra tutti i parametri disponibili per CallLegProfiles. Quando si passa il mouse su un parametro specifico, viene visualizzato un menu di scelta rapida che mostra lo scopo di ogni opzione.



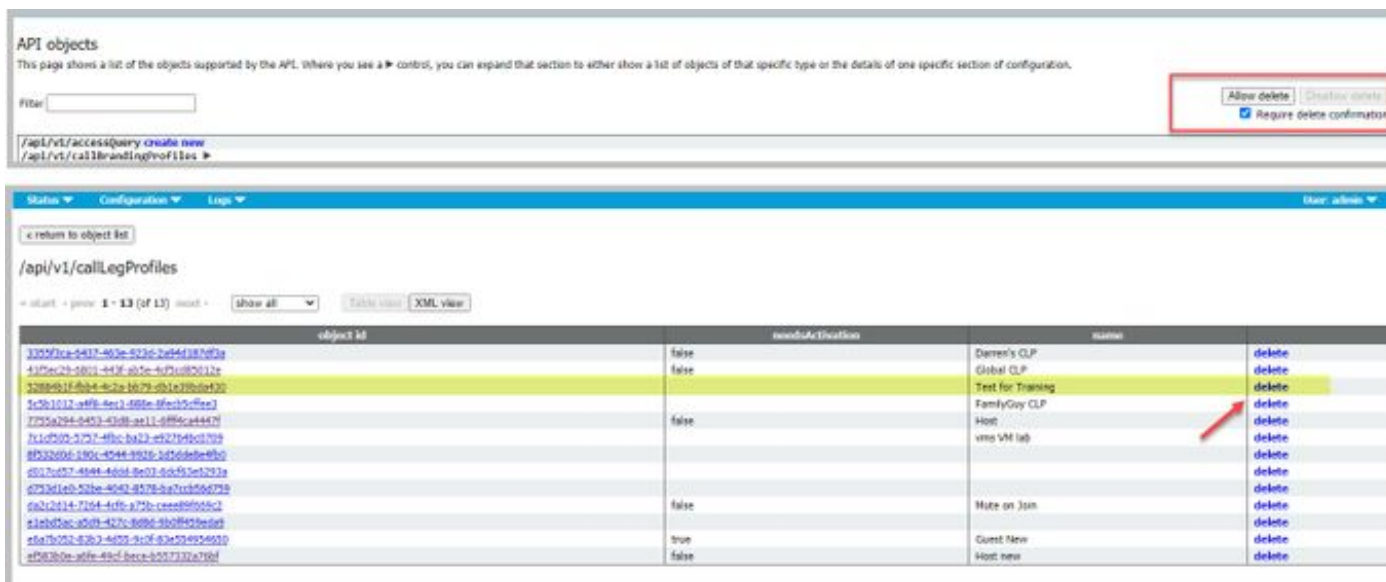
Modificare un oggetto API già creato

Quando si modificano le impostazioni di un oggetto, nella parte inferiore viene visualizzato il pulsante Modifica. Si tratta dello stesso tipo di PUT degli strumenti di terze parti.



Eliminare un oggetto API dal menu API

Per eliminare un oggetto, nella pagina Elenco oggetti principale è possibile attivare la possibilità di eliminare gli elementi. Per abilitare l'opzione da eliminare, fare clic su **Allow delete** (Consenti eliminazione), come mostrato nell'esempio:



Spiegazione delle richieste API (utilizzando strumenti di terze parti)

Le quattro richieste di base vengono spiegate tramite un esempio di configurazione.

POST HTTP

Passaggio 1. Utilizzare **POST** per creare un oggetto.

In questo esempio viene creato uno spazio CMS utilizzando questa richiesta. Per creare lo spazio tramite API, consultare la documentazione dell'API. Per questo esempio ho utilizzato la guida all'API CMS 2.4, ma è necessario utilizzare le guide all'API più recenti, disponibili [qui](#)

La sezione 6.2 contiene informazioni su come creare e modificare un coSpazio.

6.2 Creating and Modifying a coSpace	37
6.2.1 Secondary coSpace URIs	40
6.2.2 Auto-generation of coSpace callId	40

La prima frase dice che per creare uno spazio, è necessario inviare un post a /coSpaces. Quindi, dice che l'ID dello spazio sarà nell'intestazione Location del 200 OK. Bene, ora sai come creare uno spazio. È sufficiente inviare un **POST** a **https://<WebAdminIP>/api/v1/coSpaces**.

- **Creating:** POST method to the "/coSpaces" node. If the coSpace was created successfully, a "200 OK" response is received, and the "Location" header contains the ID for the new coSpace

Specificare i parametri per il **POST**.

Nella sezione 6.2 della documentazione è riportata una tabella che elenca tutti i parametri che è possibile utilizzare.

Parameters	Type/Value	Description/Notes
name	String	The human-readable name that will be shown on clients' UI for this coSpace
uri	String (URI user part)	The URI that a SIP system would use to dial in to this coSpace. (The URI "user part" is the part before any '@' character in a full URI.)
secondaryUri	String (URI user part)	The secondary URI for this coSpace - this provide the same functionality as the "uri" parameter, but allows more than one URI to be configured for a coSpace. (The URI "user part" is the part before any '@' character in a full URI.)

Ad esempio: Creare uno spazio con il nome **APITest** e una parte utente URI di **APITestURI**

Il tipo di contenuto è **application/x-www-form-urlencoded** e il contenuto è **name=APITest&uri=APITestURI**

Quando si aggiungono questi parametri, la richiesta è completa, come mostrato nell'immagine.

```
POST https://<WebAdminIP>/api/v1/coSpaces HTTP/1.1
Host: <WebAdminIP>
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive
```

name=APITest&uri=APITestURI

Nell'immagine è illustrata una risposta alla richiesta precedente.

```
HTTP/1.1 200 OK
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Location: /api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae
Vary: Accept-Encoding
Content-Encoding: gzip
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
```

Si noti l'intestazione Posizione nella risposta.

Location: /api/v1/coSpaces/**70ca0ed7-4e50-428c-b9ab-4e68faeb86ae**

70ca0ed7-4e50-428c-b9ab-4e68faeb86ae è l'ID del nuovo Spazio creato. L'ID è utile quando è necessario effettuare richieste API future che abbiano come destinazione lo stesso spazio.

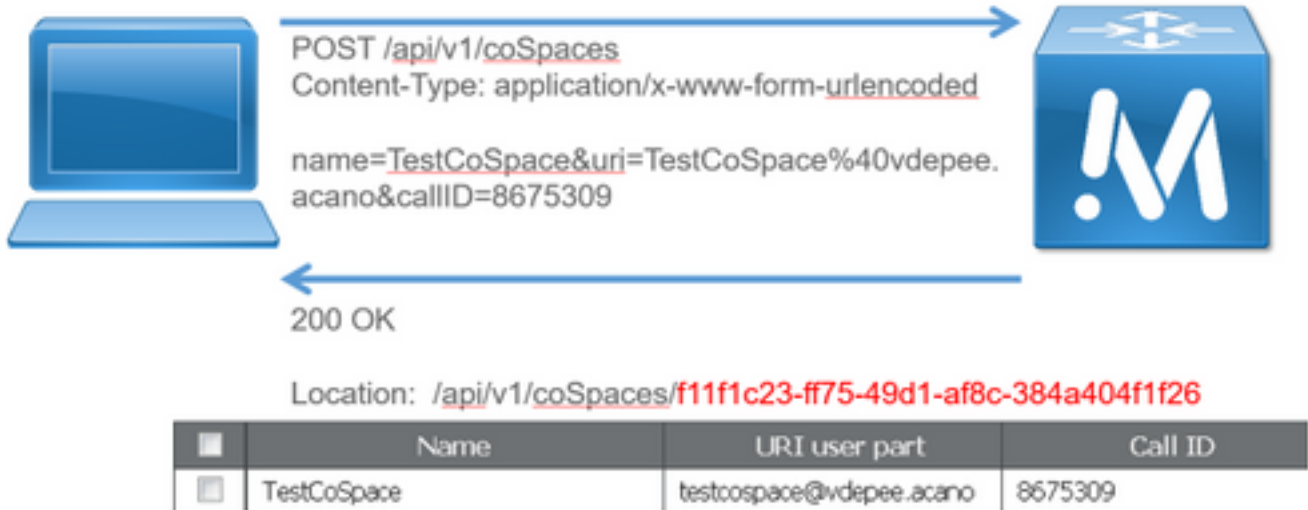
Lo spazio può essere visualizzato nel WebAdmin di CMS. Selezionare **Configurazione > Spazi**.

<input type="checkbox"/>	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
<input type="checkbox"/>	APITest	apitesturi					not set

L'immagine riassume il **POST** della richiesta.

HTTP POST

- Creates new object



HTTP GET

Passaggio 2. Dopo aver creato lo spazio, estrarre la configurazione corrispondente.

A tale scopo, utilizzare il metodo HTTP GET.

Utilizzare l'ID dello spazio creato dall'installazione Posizione. L'URL è <https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae>. Eseguire un'operazione **GET** in questa pagina.

Esempio di richiesta GET:

```
GET https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
Host: <WebAdminIP>
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Cookie: session=logout
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive
```

Risposta per la richiesta GET:

```
HTTP/1.1 200 OK
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Type: text/xml
Vary: Accept-Encoding
Content-Length: 159
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

```
<?xml version="1.0"?><coSpace id="70ca0ed7-4e50-428c-b9ab-4e68faeb86ae"><name>APITest</name><autoGenerated>>false</autoGenerated><uri>apitesturi</uri></coS
```

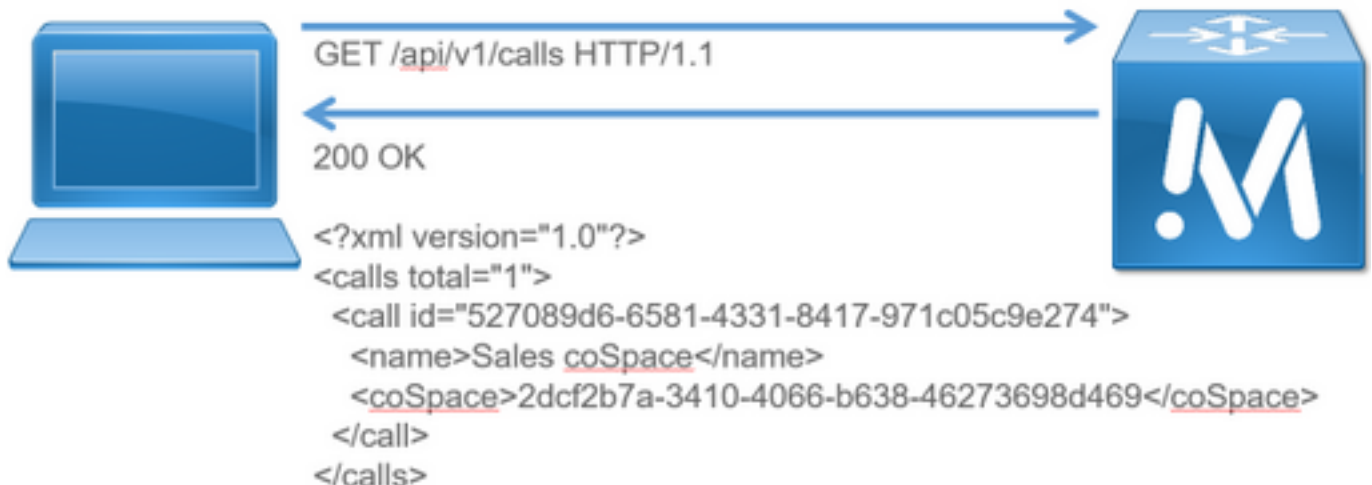
pace>

Nota: La risposta è una configurazione con codifica XML dello spazio.

L'immagine riepiloga la richiesta **GET**.

HTTP GET

- Retrieves existing information
- No Content in Body



HTTP PUT

Passaggio 3. Apportare una modifica allo spazio (se necessario).

Nell'esempio viene mostrato come modificare lo spazio creato. Si supponga che sia necessario aggiungere allo spazio una parte secondaria Utente.

Fare riferimento al documento API. Indica il parametro necessario da utilizzare, ovvero: **URI secondario**.

Aggiungere un URI di asdf. Scrivere una richiesta simile alla richiesta creata per POST.

Esempio di richiesta PUT:

```
PUT https://172.18.105.244/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
Host: 172.18.105.244
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 17
Cookie: session=b810c447daaeab6cdc6e019c
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive
```

secondaryUri=asdf

Risposta per la richiesta PUT:

```
HTTP/1.1 200 OK
Date: Tue, 12 Apr 2016 19:11:02 GMT
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 0
```

Le modifiche possono essere visualizzate in WebAdmin di CMS. Selezionare **Configurazione > Spazi**.

<input type="checkbox"/>	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
<input type="checkbox"/>	APITest	apitesturi	asdf				not set

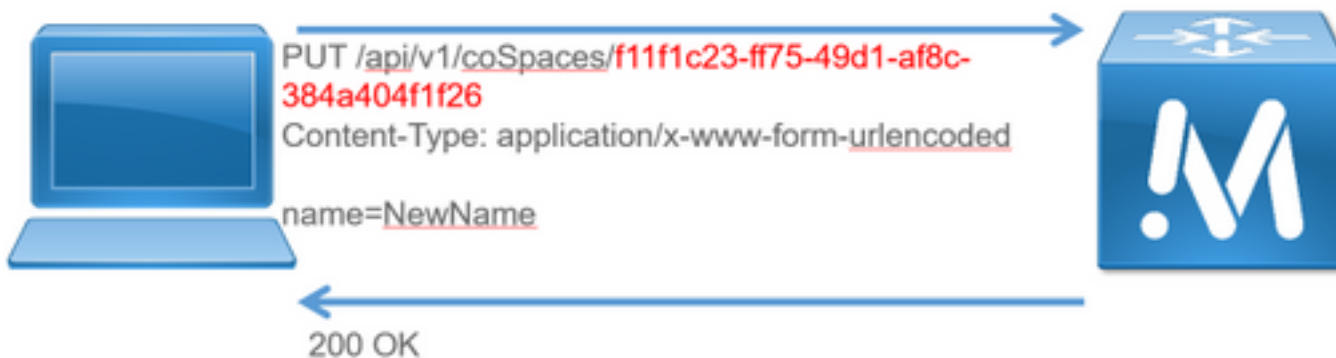
e tramite **GET**:

```
<?xml version="1.0"?><coSpace id="70ca0ed7-4e50-428c-b9ab-4e68faeb86ae"><name>APITest</name><autoGenerated>>false</autoGenerated><uri>apitesturi</uri><secondaryUri>asdf</secondaryUri></coSpace>
```

L'immagine riepiloga la richiesta **PUT**.

HTTP PUT

- Modifies existing object



<input type="checkbox"/>	Name	URI user part	Call ID
<input type="checkbox"/>	NewName	testcospace@vdepee.acano	8675309

ELIMINAZIONE HTTP

Passaggio 4. Eliminare lo spazio (se necessario).

Il metodo **DELETE** è simile al metodo **GET**.

Esempio di richiesta **DELETE**:

```
DELETE https://172.18.105.244/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
```

Host: 172.18.105.244
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Cookie: session=4d13c7ebe739b662dc6e019c
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive

Risposta per la richiesta DELETE:

HTTP/1.1 200 OK
Date: Tue, 12 Apr 2016 19:16:37 GMT
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 0

Le modifiche possono essere visualizzate in WebAdmin di CMS. Selezionare **Configurazione > Spazi**.

Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout	
						not set	<input type="button" value="Add New"/> <input type="button" value="Reset"/>

e tramite **GET**:

```
<?xml version="1.0"?><failureDetails><coSpaceDoesNotExist /></failureDetails>
```

L'immagine riepiloga la richiesta **DELETE**.

HTTP DELETE

- Destroys an object



Name	URI user part	Call ID

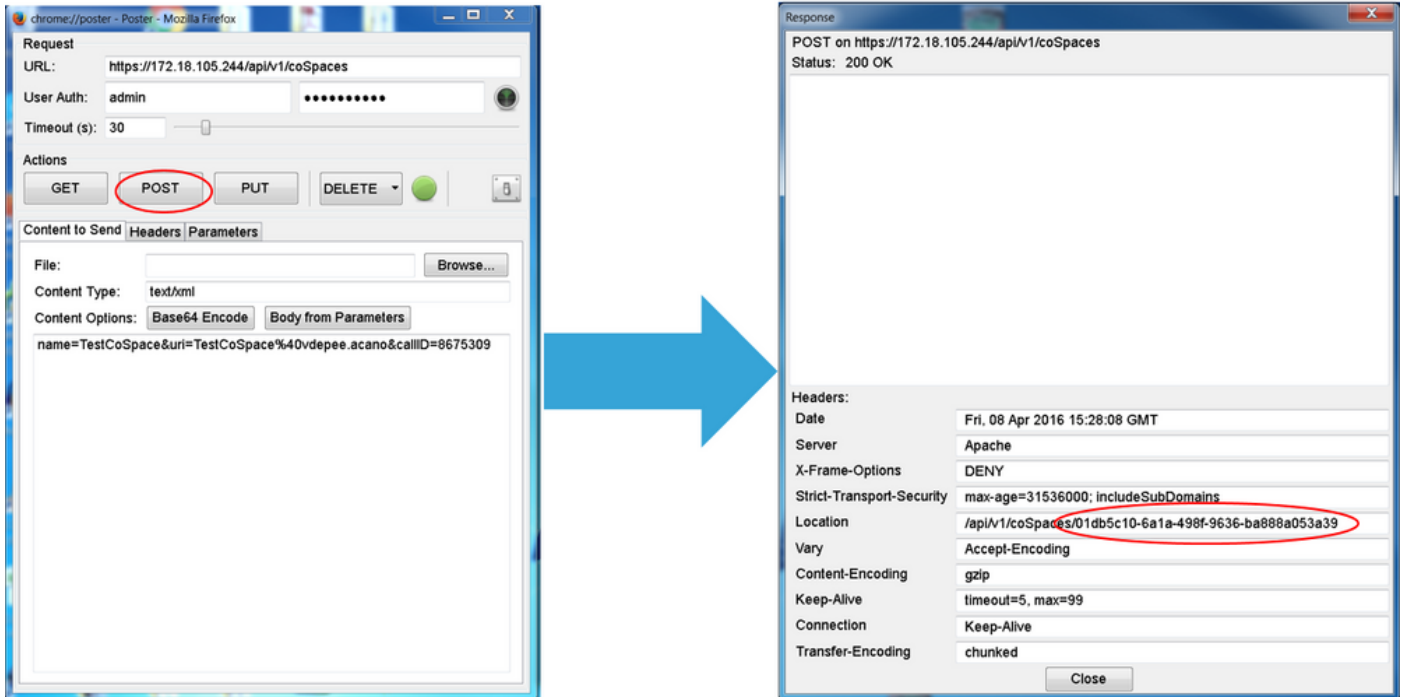
Client API

POSTER

Nella casella superiore del poster è possibile immettere l'URL per le richieste.

Nei campi Autenticazione utente è possibile immettere il nome utente e la password nell'ordine indicato. Quindi, se si sta eseguendo un'operazione **GET** o **DELETE**, scegliere i pulsanti corrispondenti. Ad esempio: fare clic su **GET** per visualizzare una finestra popup con la risposta. Per **DELETE**, verificare che **DELETE** sia selezionato e fare clic sul pulsante verde.

Poster (Firefox)



Per **POST** e **PUT**, è necessario aggiungere il contenuto. Selezionare la scheda Parametri e aggiungere i nomi e i valori dei parametri. Quindi, tornare al pulsante Contenuto da inviare e scegliere **Corpo da Parametri**.

Inviare POST e/o PUT.

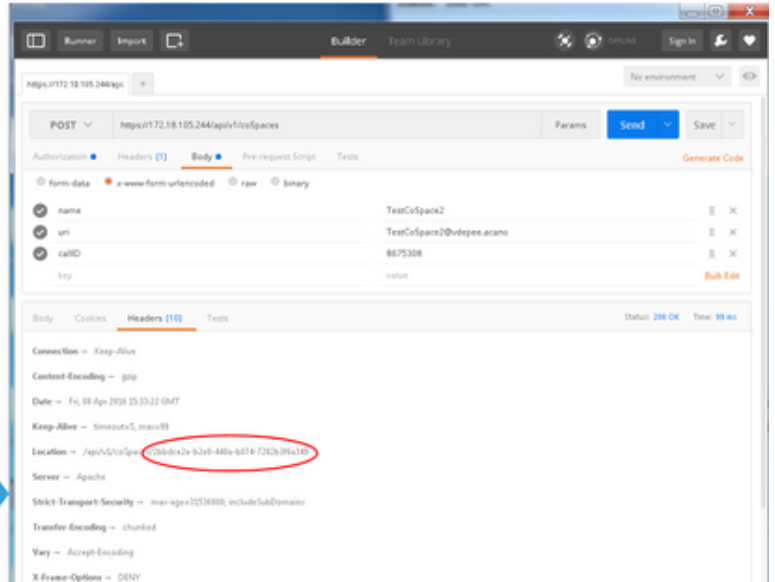
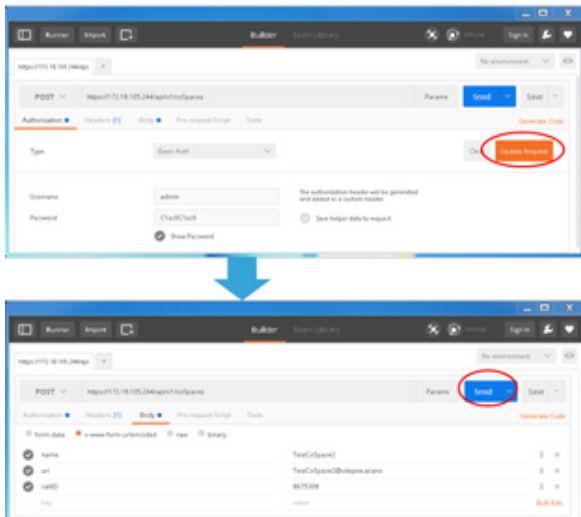
POSTINO

In Postman, in alto a sinistra, scegliere il Metodo che si desidera utilizzare dalla casella a discesa e immettere l'URL della richiesta.

Per Autorizzazione, scegliere **Basic-Auth** e immettere il nome utente e la password. Quindi, scegliere **Aggiorna richiesta**. Nella scheda Intestazioni viene visualizzata un'intestazione di autorizzazione.

Se la richiesta è di tipo POST/PUT, passare alla scheda Corpo, scegliere **x-www-form-urlencoded** e immettere i parametri e i valori. Al termine, scegliere **Invia**.

Postman (Chrome)



Verifica

Il metodo di verifica è illustrato in ciascuna richiesta.

Risoluzione dei problemi

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.